



# Transacciones

ADMINISTRACIÓN DE BASES DE DATOS

Autores: Yassin El Ghazouani Azzouzi

Rafael Cobos Sánchez

# Índice

- 1. Introducción.
  - 1.1. Propiedades.
- 2. Control de transacciones.
  - 2.1. Sentencias.
  - 2.2. COMMIT VS ROLLBACK.
- 3. Transacciones autónomas.
- 4. Transacciones distribuidas.
  - 4.1. Bases de datos distribuidas.
  - 4.2. Transaction Manager.
- 5. Conclusiones.
- 6. Bibliografía.

# 1. Introducción.

- ¿Qué es una transacción?
- Las transacciones deben cumplir 4 propiedades (ACID):
  - Atomicidad (**A**tomicity).
  - Consistencia (**C**onsistency).
  - Aislamiento (**I**solation).
  - Permanencia (**D**urability).

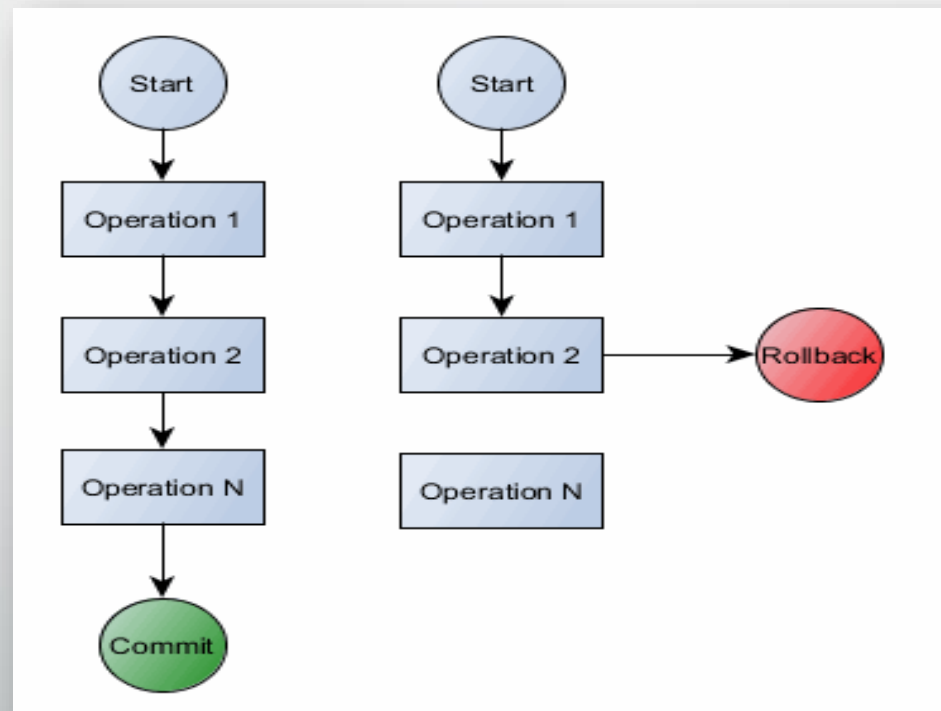


# 1.1. Propiedades ACID.

- Atomicidad: Asegurarse de que todas las operaciones se han realizado correctamente. En caso de fallo, deshacer todas las operaciones.
- Consistencia: Solo se empiezan aquellas transacciones que se puedan acabar sin romper reglas ni directrices de integridad en la base de datos.
- Aislamiento: Las transacciones no deben afectarse entre si.
- Permanencia: Los cambios de una transacción persistirán y se mantendrán en la base de datos.

## 2. Control de transacciones.

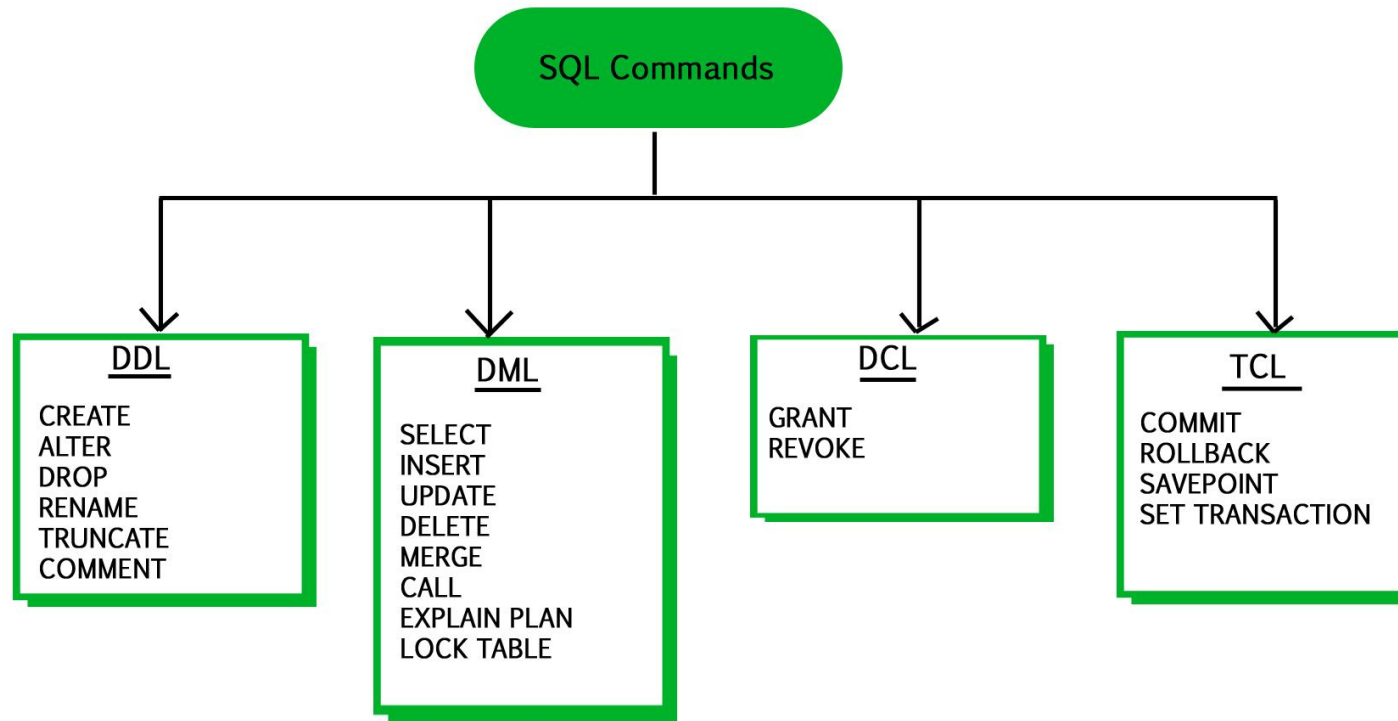
- ¿Qué es el control de transacciones?
- ¿Para qué sirve?



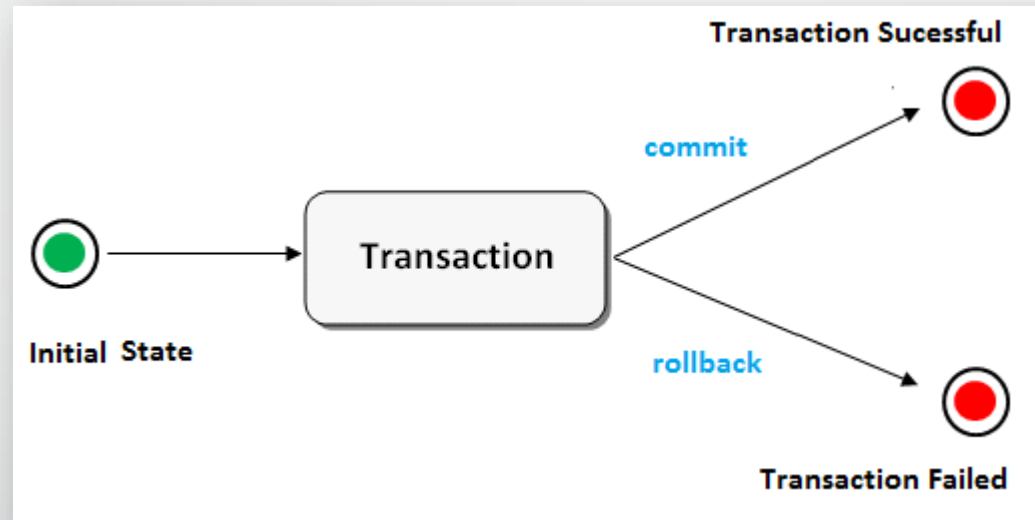
## 2.1. Sentencias.

- TCL
  - SET TRANSACTION <<NAME>>
  - COMMIT
  - ROLLBACK
  - SAVEPOINT
  - ROLLBACK TO SAVEPOINT

## 2.1. Sentencias.



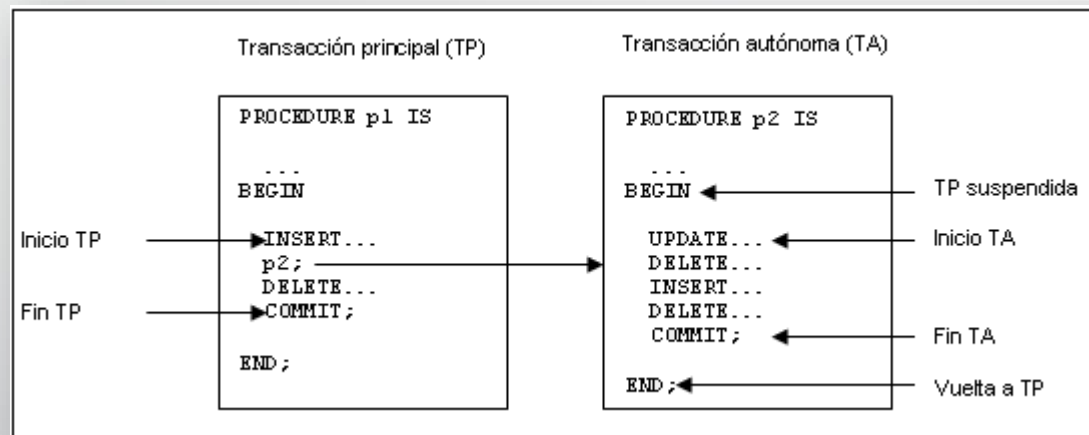
## 2.2. COMMIT VS ROLLBACK.





### 3. Transacciones autónomas.

- Las transacciones autónomas son aquellas que se ejecutan dentro de otras y son independientes de ellas. Cuando se ejecuta la transacción autónoma, se detiene la ejecución de la principal.



### 3. Transacciones autónomas.

- Las transacciones autónomas tienen su propio alcance de confirmación y reversión, esto asegura que su resultado no se verá afectado por la no confirmación de la transacción principal.
- Creación de una transacción autónoma:

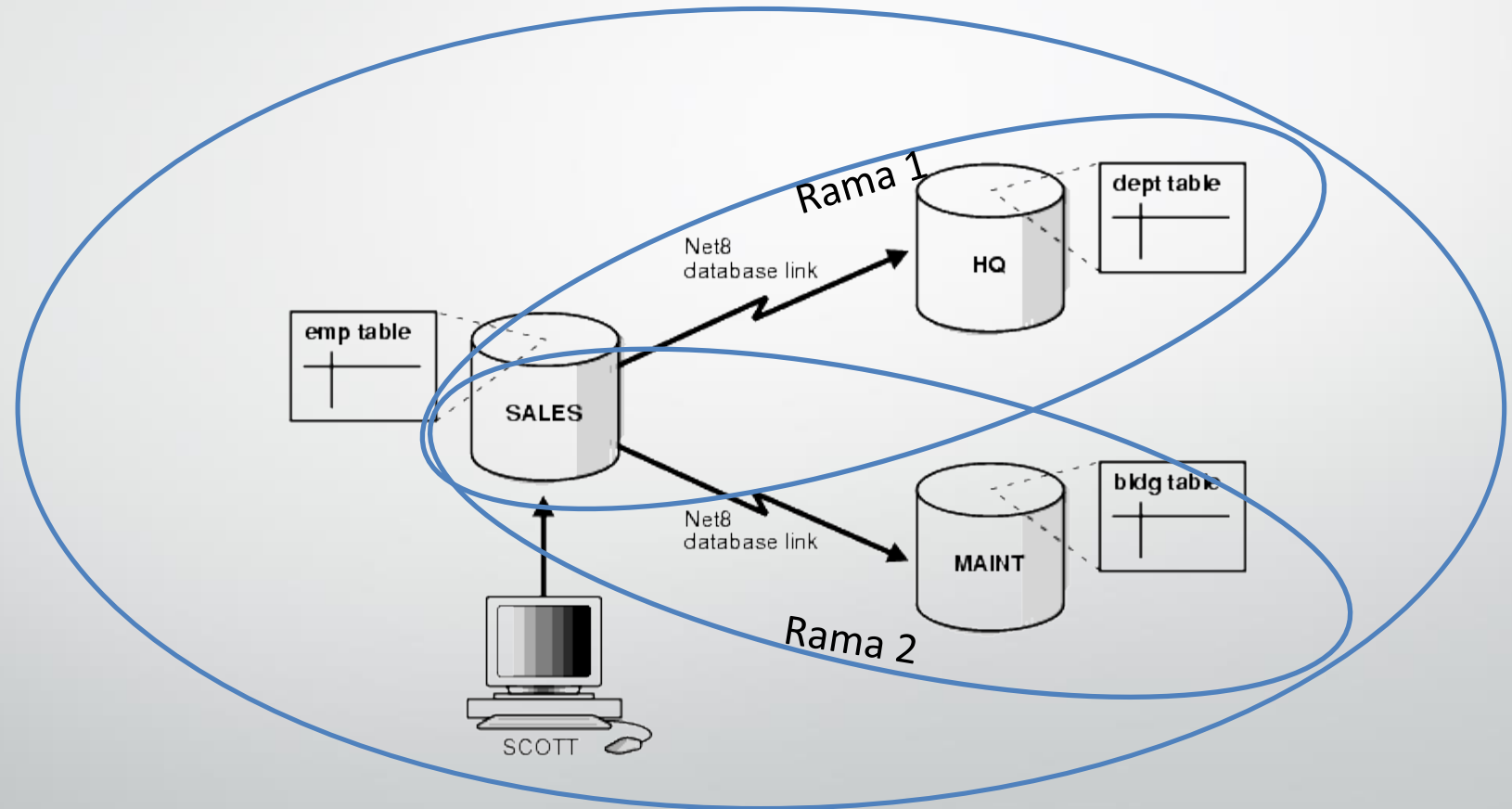
```
CREATE OR REPLACE PROCEDURE Grabar_Log(descripcion VARCHAR2)
IS
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
    INSERT INTO LOG_APLICACION
    (CO_ERROR, DESCRIPCION, FX_ERROR)
    VALUES
    (SQ_ERROR.NEXTVAL, descripcion, SYSDATE);

    COMMIT; -- Este commit solo afecta a la transaccion autonoma
END ;
```

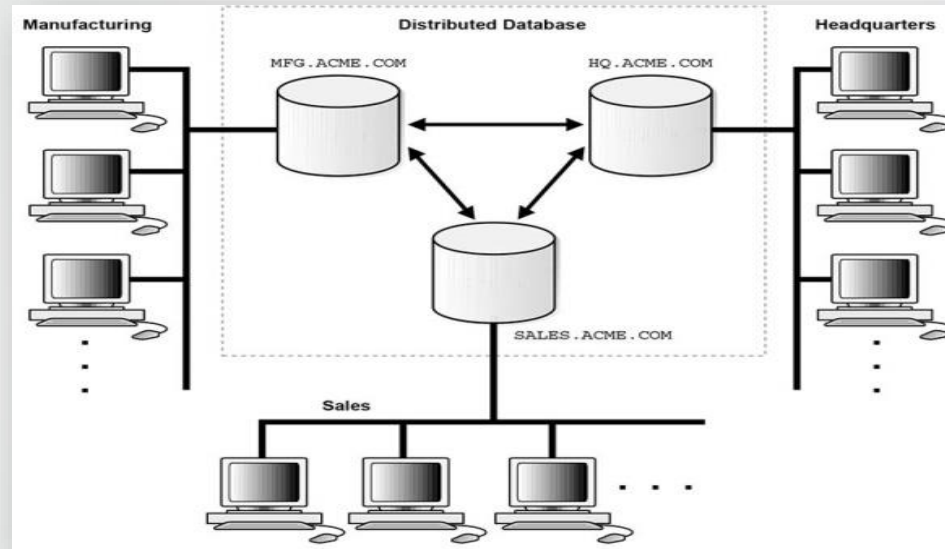
```
DECLARE
    producto PRECIOS%TYPE;
BEGIN
    producto := '100599';
    INSERT INTO PRECIOS
    (CO_PRODUCTO, PRECIO, FX_ALTA)
    VALUES
    (producto, 150, SYSDATE);
    COMMIT;
EXCEPTION
WHEN OTHERS THEN
    Grabar_Log(SQLERRM);
    ROLLBACK;
    /* Los datos grabados por "Grabar_Log" se escriben en la base
    de datos a pesar del ROLLBACK, ya que el procedimiento está
    marcado como transacción autonoma.
    */
END;
```

## 4. Transacciones distribuidas.

- ¿Qué son?



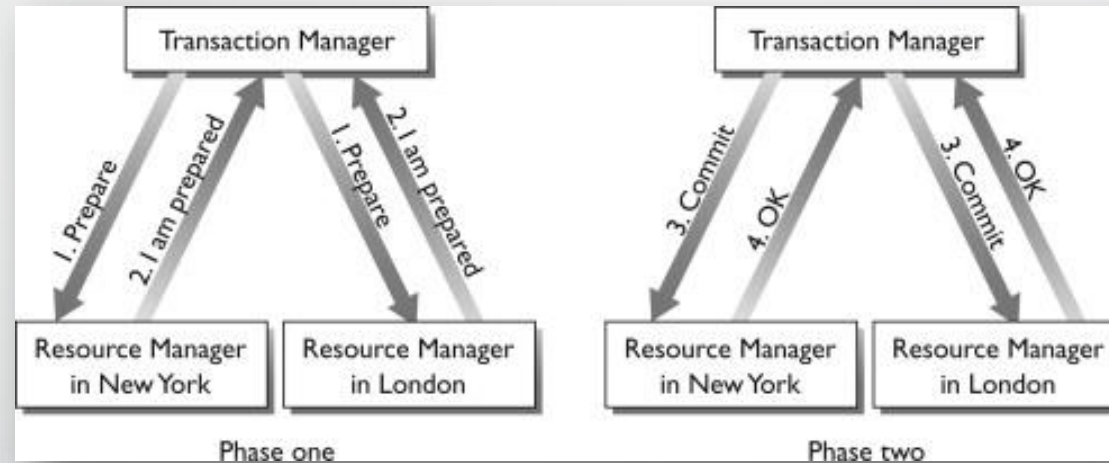
## 4.1. Bases de datos distribuidas.



- Actúan como una red.
- Cada base de datos se considera un nodo de la red.
- Se puede acceder a cualquier dato tanto en local como en remoto.
- Todas las transacciones son transparentes al usuario.

## 4.2. Transaction Manager.

- Gestor software que coordina las transacciones. Es considerado un middleware.



## 5. Conclusiones.

- Anteriormente una transacción era complicada de hacer ya que se realizaba de forma manual, es decir, haciendo transferencias de forma real de una a otra.
- Hoy en día SQL nos ofrece la forma de realizar transferencias de forma electrónica o virtual.
- Dichas operaciones son fáciles y confiables siempre y cuando se cumpla con las propiedades ACID y se manejen los comandos commit y rollback adecuadamente.

## 6. Bibliografía.

- [https://es.wikipedia.org/wiki/Transacci%C3%B3n\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Transacci%C3%B3n_(inform%C3%A1tica))
- <https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Transacciones.aspx>
- <https://www.ediciones-eni.com/open/mediabook.aspx?idR=b320e75c216cf83a0324a4ca458bc1eo>
- <http://www.devjoker.com/contenidos/catss/64/Transacciones-autonomas.aspx>